An Evolvable Artificial Chemistry Featuring Continuous Physics and Discrete Reactions

Thomas E. Portegys

Illinois State University
Campus Box 5150
Normal, Illinois, USA 61790-5150
portegys@ilstu.edu

Abstract

This paper describes an artificial chemistry featuring atoms and molecules moving and colliding in a continuous manner in a viscous fluid filling a 2D cellular space. Chemical reactions are mappings of discrete cellular configurations to parameterized actions on atoms. Actions allow atom creation and destruction, bonding and unbonding to make and break molecules, orientation, type change, and propulsion. Actions are easily added in this extensible model. An example involving a complex "foraging" reaction is provided as a demonstration of the capabilities of the framework. The reaction rules can be evolved by a genetic algorithm to exhibit a desired set of reactions. A portion of the foraging reaction was evolved to demonstrate this.

Introduction

This work was done in the spirit of the growing belief that artificial life requires a sufficiently rich physics and chemistry in which to develop (Bedau et al. 2000, Dittrich et al. 2001). The attempt here is to combine a simple continuous Newtonian particle physics with the well-known and readily computable cellular automaton (CA) model as a chemistry implementation. There are two additional purposes for using a CA: (1) the chemistry of real organic molecules, involving complex foldings and bondings, is computationally infeasible (Zagrovic et al. 2002); and (2) it serves the aim of artificial life research to discover underlying mechanisms necessary for the existence of living systems.

This project was primarily inspired by three recent efforts: Hutton's artificial chemistry Squirm3 (2002), Smith, Turney and Ewaschuk's JohnnyVon (2003), and my prior work on intercellular signaling (Portegys 2002). Prior to these, Dittrich et al. (2001) compared a wide range of artificial chemistry approaches, including assembler automata (Rasmussen et al. 1990, Ray 1992, Adami and Brown 1994), Ono and Ikegami's autocatalytic membrane formation (1999), and lattice molecular systems (McMullin and Varela 1997), in which the atoms comprising a molecule map discretely to cellular space.

Squirm3 is a lattice molecular system in which mobile molecules self-replicate using available atoms in a 2D cellular space. Atoms have a fixed type and a variable state. Chemical reaction rules based on type, state and proximity determine the states and bonding status of atoms. Atoms move by jumping from cell to cell through simulated Brownian motion; however, atoms bound into molecules are largely immobile. Beginning with a soup of inert atoms exposed to state-disrupting cosmic rays, self-replication was shown to spontaneously occur.

In JohnnyVon, by contrast, T-shaped atoms called codons move through a continuous 2D space and interact exclusively through force fields; collisions do not occur. There are two types of codons, distinguished by their field "colors". Each codon is an automaton containing a set of rules governing its field strengths in response to signals represented by the proximity of other codon fields.

This project uses the intercellular signaling scheme put forth in my recent work addressing the problem of allowing cells in a CA to communicate without disturbing the state of intervening cells. In this project, signals are chiefly directed toward atoms residing within cells, rather than the cells themselves.

Description

This system is an artificial chemistry featuring atoms and molecules moving and colliding in a continuous manner in a viscous fluid filling a 2D cellular space. A molecule consists of a bound set of atoms. Chemical reactions are mappings of discrete cellular configurations to parameterized actions on atoms. Actions allow atom creation and destruction, bonding and unbonding to make and break molecules, orientation, type change, and propulsion. Time proceeds in discrete steps.

Atoms and molecules

Atoms are elementary particles possessing a type, mass, radius, charge, orientation, position, and velocity. A molecule is a set of atoms connected by bonds. The atoms within a molecule must form a connected set; there must be a bond path directly or indirectly connecting any pair of atoms. A bond may be of variable but limited length.

Forces resulting from elastic collisions and chemical reactions result in continuous velocity and position updates according to Newtonian physics. For the initial implementation, rotational momentum is not supported. The atoms within a molecule respond to forces as a single unit; so a collision of one of its atoms results in a force applied to the mass of the entire molecule. An atom is oriented in one of the eight compass directions. It may also be in a mirrored state to support symmetric reactions.

Space

The space is a 2D cellular grid filled with a viscous fluid that impedes the movement of atoms through it. In the current implementation, the diameter of an atom is equal to the dimensions of a cell.

Reactions

Reactions arise from the interaction of atoms. At any moment, the center of an atom resides within a unique cell. This cell forms the center of a 3x3 Moore neighborhood of cells. A reaction specifies a configuration of atom types residing within a neighborhood. Since atoms have an orientation, their cell neighborhoods are oriented accordingly, as shown in Figure 1.

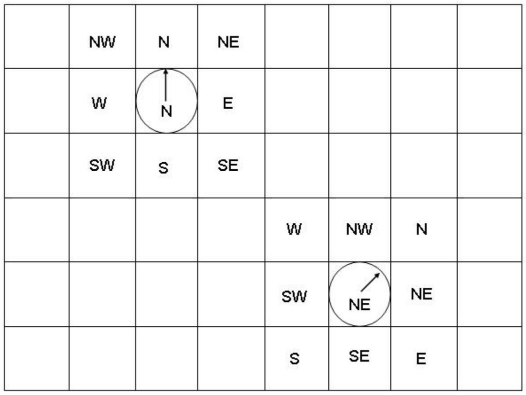


Figure 1: Oriented Cell Neighborhoods

A neighborhood cell value is one of {atom type, empty, occupied, ignore}. The neighborhood evaluation is a conjunction of all 9 value matches. An empty value specifies that a cell must not contain any atom. This could be used for inhibitory control of a reaction. The occupied value matches any atom type. The ignore value positively matches any cell condition.

A reaction consists of the following: {neighborhood, action, target, parameters}. Target is the location of the cell where the action is directed via a signal. This may reside outside of the neighborhood. Parameters apply to actions. These are: {atom type, orientation, strength, tendency, delay, duration}. The actions are:

- Create and bond atom. Parameters: type and orientation of created atom.
- Destroy atom. Parameter: target atom type.
- Bond/unbond acting atom to/from atom. Parameter: target atom type.
- Grapple atom: bond to and move atom to location in acting atom's neighborhood. Parameters: target atom type, orientation relative to neighborhood center cell.
- Orient atom relative to acting atom's orientation. For example, if acting atom is oriented east, and it orients an atom to its west, the atom will acquire a north orientation. Parameters: target atom type and orientation.
- Modify atom type. Parameter: target atom old and new types.
- Propel atom: apply a propulsion force to a specific atom (see below).

Propel atom action:
This is a reaction in which atoms apply propulsion forces to themselves and other atoms, which when combined with the viscous fluid medium gives the chemistry a more active and interesting nature, albeit at the expense of real world fidelity. More natural alternatives would be to employ charge forces or Brownian motion to achieve coarser and slower movements and reactions. The parameters for the propel action are: target atom type, force direction (orientation), speed (strength), tendency (relative probability), delay, and duration. Since the atoms within a molecule move as a unit, the propulsion forces are combined at a molecular level. Propulsion forces are accumulated with collision forces to update the positions and velocities of molecules.

Update cycle

1. Update positions and velocities of molecules based on accumulated forces.
2. Clear forces.
3. Compute new collision forces.
4. Match reactions to atom neighborhoods and distribute action signals to target cells.
5. Execute actions.

Results

This section presents a demonstration of the capabilities of the framework. Both Squirm3 and JohnnyVon solved the challenge of self-replication. For the sake of variety, a different problem is ventured here that I propose is of comparable complexity. The problem involves an orchestration of cooperating reactions that allows a particular molecule to randomly encounter and systematically destroy atoms of a particular type. The overall impression is one of “foraging”, a metaphor that will be used in the following description.

The molecule, called “Maxwell” (for whimsical reasons) is shown in Figure 2.

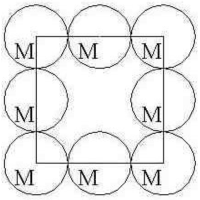


Figure 2: The Maxwell Molecule

This molecule is a ring of bonded atoms. Its atoms are labeled with ‘M’ in order to distinguish atoms “belonging” to Maxwell from others. However, the corner atoms are of a different type than the side atoms. In addition, the orientations of all the atoms point directly away from the center, so for example the top center atom is oriented north, the top right atom northeast, etc. This allows the same reaction rules to work regardless of the molecule’s relative position to external atoms. One set of reactions is structured to allow the Maxwell molecule to destroy “food” atoms by “ingesting” them. In doing so, a variety of reaction types are brought into play.

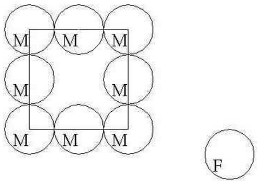


Figure 3: Beginning Ingestion Reaction

Figure 3 show the beginning of the ingestion reaction sequence as Maxwell approaches a food atom.

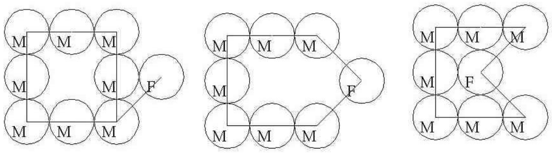


Figure 4: Ingestion Reaction Continued

Figure 4 show the next three steps. Proceeding from left to right: the lower right corner atom grapples the food atom, moving it counterclockwise over the side atom; the side atom has destroyed itself to create a pathway into the center, and the other nearby corner atom has bonded to the food atom; the food atom is grappled into the center.

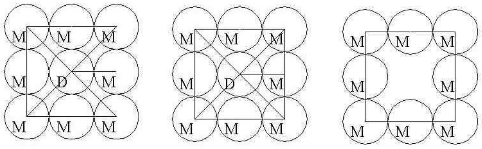


Figure 5: Ingestion Reaction Completion

The reaction completes in Figure 5 with the restoration of the side atom and rebonding of the molecule. The process is facilitated by a type change to the center food atom before it is destroyed. There are 21 reactions involved in the entire process.

Maxwell will systematically ingest food atoms it comes into contact with. The propel reaction type allows it to move about, or “forage”, in a space that is stocked with variable-sized patches of food atoms.

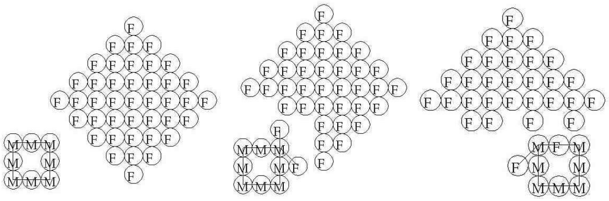


Figure 6: Foraging Reaction

Figure 6 shows Maxwell ingesting a food patch it has come into contact with. In the “programmed” version of foraging, it will move in a spiral counter-clockwise pattern around the perimeter of the patch, usually resulting in the complete ingestion of it. Figure 7 shows the finale of the process.

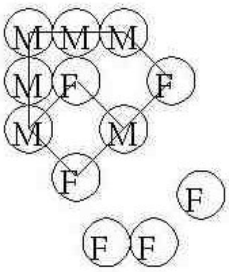
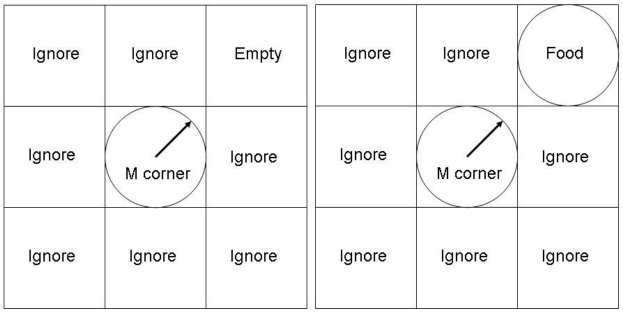


Figure 7: Foraging Reaction Finale



Ignore	Ignore	Empty
Ignore	M corner	Ignore
Ignore	Ignore	Ignore

Ignore	Ignore	Food
Ignore	M corner	Ignore
Ignore	Ignore	Ignore

Figure 8: Foraging Reaction Neighborhoods

Three propel reactions are involved in foraging. The left portion of Figure 8 shows the neighborhood of the first propel reaction that produces movement in free space (no food present). The neighborhood specifies a corner atom with an empty cell directly “north” of it. Table 1 shows the other attributes of this reaction. The direction and strength values translate to a fairly high rate of speed moving the entire molecule in a northeast direction. The tendency value indicates that, in the absence of other propel reactions, there is a 10% probability of this reaction occurring. This is important considering that this reaction applies to all four corners of the molecule. If all "fired" at every step, the propulsions would cancel and the molecule would not move. As it is, the result is a random Brownian-type motion with fairly long legs.

Attribute	Value
target type	corner
displacement	0,0
direction	north
strength	2.0
tendency	0.1
delay	0
duration	1

Table 1: First Foraging Reaction

The neighborhood in the right portion of Figure 8 applies to the second and third reactions. The second reaction attribute values are given in Table 2. This reaction moves a corner atom east when a food atom is north of it, which results in the molecule moving along the side of a food patch. The strength value causes it to move slowly, lest the food be knocked away by impact.

Attribute	Value
target type	corner
displacement	0,0
direction	east
strength	0.1
tendency	5.0
delay	0
duration	20

Table 2: Second Foraging Reaction

The tendency and duration of this reaction causes it to heavily outweigh and outlast the first reaction, so that an intermittent exposure to free space will not result in the molecule darting away from the food patch.

Attribute	Value
target type	corner
displacement	0,0
direction	north
strength	0.1
tendency	3.0
delay	20
duration	50

Table 3: Third Foraging Reaction

The third reaction is given in Table 3. The purpose of this reaction is to steer the molecule around food patch corners. The delay and duration values allow this reaction to remain in effect for a relatively long time after running off the edge of the patch. It will subsequently cause the atom to move north, resulting in the corner atom counter-clockwise from it to "engage" the next edge using the second reaction.

Evolving the chemistry

In the previous discussion, the foraging reactions were programmed to perform suitably. Two additional experiments were done using a genetic algorithm to evolve the three reactions pertaining to foraging. For the evolution procedure, a population member consisted of a set of foraging reactions. The population size was twenty, with the ten fittest members selected out of each generation as measured by foraging success. Foraging success is defined by the number of food atoms ingested times a value of ten each. A set of eight randomly selected fit members were mutated, the nature of which depended on the experiment (see below). An additional two fit members were selected for mating. Mating involved randomly selecting one reaction from each of the three parental pairs of foraging reactions.

In the first experiment (hill-climb), the action, neighborhood, target type, and displacement attributes of the reactions were fixed at the programmed values, while the direction, strength, delay, and duration attributes were set to minimum values and subjected to a simple hill-climbing mutation procedure. The aim of this experiment was to "tune" the propulsion parameters.

In the second experiment (random), the reactions were initially rendered inert by clearing them to minimum values and setting the action to a null value. Mutation consisted of randomly setting the various attributes to values within set tolerances. The exception to this was that when the action was mutated, the rest of the attributes were again cleared.

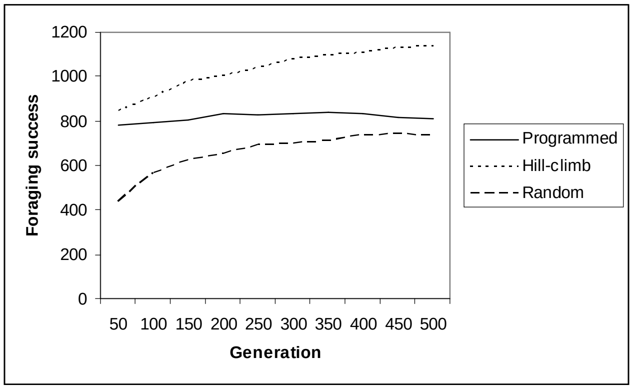


Figure 9: Foraging Evolution

Figure 9 shows the results of the experiments. The programmed (unevolved) version was run for the same number of generations and is shown as a control. The hill-climbing experiment proved to be an immediate success. Upon closer examination of the fittest members, it became obvious that the programmed foraging speed was too slow, especially the reactions controlling speed in the presence of food atoms. The random experiment quickly produced a functional set of reactions, but failed to improve dramatically after that. An inspection of the reactions of the fittest members revealed a basic propel reaction triggered by the presence of free space. Surprisingly, this was enough to approach the performance of the programmed version.

Conclusions

One of the aims of this project was to find a way to acquire some of the benefits of a continuous physical medium, such as that in which organic chemistry takes place, and also retain the computability benefits of a cellular automaton as a chemistry framework. The Maxwell molecule was designed to showcase some of the capabilities of this approach. The framework is easily extensible: new reaction types can be plugged in with few code changes.

This model has some obvious similarities and differences with Squirm3 and JohnnyVon; the former is CA based, and latter is based on continuous fields and movements. At this point it would be premature to attempt more definitive comparisons, since all are in early stages of development.

Besides further applications, such replicating molecules and semi-permeable membranes, there are a number of relatively easy enhancements that can be made to make the model more realistic, such as going to 3D (the physics code already supports this), adding rotation/angular momentum effects, and introducing charge as a source of attraction and repulsion force.

The open source C++ code is available at www.itk.ilstu.edu/faculty/portegys/research.html

References

Adami, C.; and Brown, C.T. 1994. Evolutionary learning in the 2D artificial life system avida. In Brooks, R.A. and Maes, P., editors, Artificial Life IV, Cambridge MA, MIT Press, 377-381.

Dittrich, P.; Ziegler, J.; and Banzhaf, W. 2001. Artificial chemistries – A review. Artificial Life 7:2245-275

Hutton, T. J. 2002. Evolvable Self-Replicating Molecules in an Artificial Chemistry. Artificial Life 8(4):341-356.

McMullin, B. and Varela, F.J. 1997. Rediscovering computational autopoiesis. In Husbands, P. and I. Harvey, editors, Fourth European Conference on Artifcial Life, pages 38-47, Cambridge, MA, MIT Press.

Ono, N. and Ikegami, T. 1999. Model of self-replicating cell capable of self-maintenance. In Floreano, D., J.-D. Nicoud, and F. Mondana, editors, Proc. Fifth European Conference on Artifcial Life (ECAL' 99), pages 399-406, Berlin, Springer.

Portegys, T. E. 2002. An Abstraction of Intercellular Communication", in Standish, R.K., Bedau M.A. and Abbass, H.A., editors, Artificial Life VIII Proceedings, Cambridge, MA, MIT Press, 75-78.

Rasmussen, S., C. Knudsen, R. Feldberg, and M. Hindsholm. 1990. The coreworld: Emergence and evolution of cooperative structures in a computational chemistry. Physica D, 42:111-134.

Ray, T. S. An approach to the synthesis of life. In Langton, C. G., C. Taylor, J. D. Farmer, and S. Rasmussen, editors, Artifcial Life II, pages 371-408, Redwood City, CA, Addison-Wesley.

Smith, A.; Turney, P.; and Ewaschuk, R. 2003. Evolvable Self-Replicating Molecules in an Artificial Chemistry. Artificial Life 9(1):21-40.

Zagrovic, B.; Snow, C.D.; Shirts, M.R.; and Pande, V.S. 2002. Simulation of Folding of a Small Alpha-helical Protein in Atomistic Detail using Worldwidedistributed Computing. Journal of Molecular Biology.